組み合わせ回路におけるソフトエラー伝搬率計算手法の評価

赤峰, 悠介
九州大学

原田, 翔次
九州大学

松永, 裕介
九州大学

出版情報：電気関係学会九州支部連合大会. 61, 2008-09-25
バージョン：
権利関係：

組み合わせ回路におけるソフトエラー伝搬率計算手法の評価

赤峰 悠介　原田 翔次　松永 裕介
九州大学

1 はじめに

近年，中性子によるソフトエラーに対する，LSI の耐性が低下している．極めて低い確率であるが，トランジスタへの中性子の衝突により，回路の一時的な誤動作を引き起こすことがある．微細化に伴うトランジスタの臨界電荷量の低下により，ソフトエラーの問題は深刻化している．メモリ回路においては比較的容易で低コストな対策が提案されているのに対し，論理回路のソフトエラーは十分な対策が確立されていない [1].

組み合わせ論理回路においては，中性子の衝突により論理ゲートの出力に発生したパルスが，外部出力や記憶素子に伝搬し値として取り込まれることで，回路の挙動に影響を与える．パルスを伝搬させなければ回路としてのエラーにはならないため，ソフトエラー対策として，回路の冗長化によりソフトエラー伝搬率を低減する手法が提案されている．ソフトエラー伝搬率とは，論理ゲートで発生したパルスが外部出力または記憶素子に伝搬する確率である．多くの対策は面積のオーバーヘッドを生じるため，面積の増加に対してソフトエラー伝搬率をどの程度削減できるのかが重要である．また，回路のどの部分に対策を施すのかを判断する際にもソフトエラー伝搬率は用いられる．そのため，ソフトエラー伝搬率の計算が必要である．

ソフトエラー伝搬率の計算手法として，故障シミュレーション [2] に基づく手法が提案されている．この手法は，ソフトエラー伝搬率の正確な計算ができるが，計算時間に問題がある．そのため，ソフトエラー伝搬率の計算を高速に行うための近似手法 [3] が提案されているが，計算の精度に問題がある．本稿では，正確な手法と近似手法との，計算精度と実行時間の比較評価を行う．

2 ソフトエラー伝搬率計算手法

論理ゲートで発生したパルスは，いくつかの要因によってその伝搬が阻害される．阻害要因の一つである Logic Masking は，論理的にエラーの伝搬が阻害されることである．例えば，2 入力 AND ゲートにおいて，一方の入力 i の値が 0 の場合，他方の入力 j の値はゲートの出力値に影響を与えない．このとき，j からのパルスの伝搬は阻害される．本稿では，Logic Masking による伝搬阻害のみを考慮する．

ゲート g においてパルスが発生し，外部出力，もしくは記憶素子の入力となるゲートに伝搬する確率は，g でのソフトエラーの生起確率とソフトエラー伝搬率の積であり，式 (1) で表わされる．

$$P_{err}(g) = P_{se}(g) \cdot P_{prop}(g) \tag{1}$$

式 (1) において，$P_{se}(g)$ はゲート g でのソフトエラーの生起確率，$P_{prop}(g)$ はソフトエラー伝搬率である．以降は，外部出力，もしくは記憶素子の入力となるゲートを PO(Primary Output) と呼ぶ．$P_{err}(g)$ が大きいゲートは，ソフトエラー対策を施す必要性の高い箇所であるといえる．また，対策の効果を測るために，回路全体のエラー率が用いられる．回路のエラー率は，式 (2) で表わされる．

$$P_{cer} = \sum_{g \in G} P_{err}(g) \tag{2}$$

式 (2) において，G は全ゲートの集合を表す．P_{cer} の小さい回路は，ソフトエラー耐性が高いといえる．

ソフトエラー伝搬率の計算手法として，故障シミュレーションに基づく手法，mask bit method を紹介する．

2.1 故障シミュレーションに基づく手法

パルスの発生は，一時的な論理値の反転である．そのため，縮退故障を対象としたテストパターン生成に用いられる故障シミュレーション [2] を，ソフトエラー伝搬率の計算に応用することができる．

ゲート g でのパルスの発生を仮定した回路を故障回路とする．正常回路と故障回路に同じ入力パターンを与えて論理シミュレーションを行い，出力値を比較する．出力値が異なる場合，パルスが回路の PO に伝搬することを意味する．ただし，考えられる全ての入力パターンに対して行うことは現実的には不可能であり，ランダムサンプリングで入力パターン列を決定するのが一般的である．各入力パターンに対するエラー伝搬の有無を検査して伝搬する回数をカウントし，この回数を $count(g)$ とする．ソフトエラー伝搬率 P_{fs} は，$count(g)$ と入力パターン数 k を用いて，以下の式で表わされる．

$$P_{fs}(g) = count(g)/k \tag{3}$$

故障シミュレーションに基づく手法は，与えた入力パターン列に対する正確なソフトエラー伝搬率を求めることができる．しかし，全ゲートのソフトエラー伝搬率を求めるためには，各ゲートでのソフトエラー発生を仮定した故障回路に対して論理シミュレーションを行う必要があり，実行時間は最悪，ゲート数の 2 乗に比例する．

2.2 mask bit method

mask bit method[3] は，入力パターン列に対して，一度論理シミュレーションを行い，結果をもとに，PO 側のゲートからエラー伝搬の有無を検査する手法である．

まず，入力パターン列に対して論理シミュレーションを行い，各ゲートにおいて，出力値をビット列として保持しておく．各入力パターンに対するゲート g の値が PO に伝搬するかを表すビット列を $mask(g)$ とする．伝搬するビットは 1，されないビットは 0 で表わされる．$mask$ の計算は出力側のゲートから行う．値が PO に伝搬するとき 1 であるので，PO のゲートの $mask$ は，全ビットが 1 である．図 1 の回路を対象にして，$mask(g_1)$ の計算例を述べる．g_1 の出力値が g_4 に伝搬するかを表すビット列を，

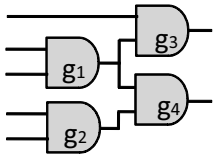

図 1: 対象とする回路

$l\_mask(g_1, g_4)$ とする．また，g_1 の出力値が，g_4 を経由して PO へ伝搬するかを表すビット列を，$g\_mask(g_1, g_4)$ とする．g_4 は AND ゲートであるので，g_2 が 1 のとき，g_1

の値が伝搬する．よって，$l\_mask(g_1, g_4)$ は，式 (4) で表わされる．

$$l\_mask(g_1, g_4) = sig(g_2) \tag{4}$$

g_1 の値が g_4 を経由して PO に伝搬するのは，g_1 から g_4 に伝搬してかつ g_4 から PO に伝搬するときである．よって，$g\_mask(g_1, g_4)$ は，式 (5) で求める．

$$g\_mask(g_1, g_4) = l\_mask(g_1, g_4)\ \&\ mask(g_4) \tag{5}$$

g_1 のエラーが PO に伝搬するのは，g_3 を経由して伝搬するもしくは g_4 を経由して伝搬するときである．よって，$mask(g_1)$ は，以下の式で表わされる．

$$mask(g_1) = g\_mask(g_1, g_3)\ |\ g\_mask(g_1, g_4) \tag{6}$$

ゲート g_1 のソフトエラー伝搬率を，式 (7) で求める．

$$P_{mb}(g_1) = ones(mask(g_1))/k \tag{7}$$

mask bit method は，一度の論理シミュレーションと，出力側からの mask の計算により，各ゲートのソフトエラー伝搬率を求める．実行時間はゲート数に比例し，故障シミュレーションと比較して高速である．ただし，ゲートのある入力値のエラー伝搬の有無を判定する際に，他の入力値が正常値であると仮定している．そのため，ゲートの複数の入力にエラーが伝搬する場合を考慮しておらず，再収れんパスを持つ回路では，ソフトエラー伝搬率が正確でない可能性がある．

3 実験

故障シミュレーションに基づく手法，mask bit method によりソフトエラー伝搬率を求め，mask bit method の計算精度の評価および実行時間の比較を行った．計算精度の評価は，故障シミュレーションに基づく手法に対する誤差の最大値と，ソフトエラー伝搬率の総和を用いる．誤差の最大値は以下の式で求める．

$$e_{max} = \max_{g \in G} |P_{mb}(g) - P_{fs}(g)| \tag{8}$$

また，ソフトエラー伝搬率の総和は，式 (1)(2) より，各ゲートでのソフトエラーの生起確率を同一とみなしたときの回路のエラー率と比例するため，これを評価に用いる．

C++でプログラムを作成し実験を行った．実験に用いたマシンの CPU は IntelXeon3.0GHz，メモリは 16GB である．ベンチマーク回路として，ITC'99 ベンチマークセットのうち回路規模の大きい 12 個の回路を，2 入力 NAND ゲートと NOT ゲートのみからなる回路に変換したものを用いた．表 1 に実験結果を示す．表 1 において，fault sim, mask bit はそれぞれ，故障シミュレーションに基づく手法，mask bit method を表し，sum はソフトエラー伝搬率の総和を表す．time は実行時間である．また，mask/fault は総和の比であり，e_{max} は誤差の最大値である．ソフトエラー伝搬率の誤差の最大値は，全てのベンチマークで 30%以上であり，最大である b15 では 62.1%である．一方，ソフトエラー伝搬率の総和は，二つの手法間でほとんど差が見られない．また，故障シミュレーションの実行時間は，mask bit method と比較して，50 倍〜70 倍程度である．

ソフトエラー伝搬率のゲート毎の誤差の最大値は大きいが，総和はほとんど差がないといえる．より詳細な解析として，図 2 に，b15 のゲート毎の誤差の分布を示す．図 2 において，横軸は誤差の範囲，縦軸は範囲内のゲート数である．例えば，誤差が 10%より大きく 20%以下であるゲート数は，58 である．一部のゲートにおいては大きな誤差があり正負両方に分布が見られる．しかし，多くのゲートにおいて誤差はほとんどない．他のベンチマーク回路においても同様の傾向があることがわかった．ソフトエラー伝搬率の総和に差が現れない原因として，誤差が大きいゲートが少ないこと，誤差が正負両方に分布していることが考えられる．

以上より，mask bit method は，回路のエラー率の計算としては正確であり，回路の評価に用いるには妥当であると考えられる．しかし，ゲート毎のソフトエラー伝搬率は，一部のゲートにおいて大きな誤差を生じており，対策を施すべき箇所の特定に用いるのは適切でないと考えられる．

表 1: 実験結果

	fault sim		mask bit		mask/	e_{max}
	sum	time[s]	sum	time[s]	fault	
b14	1862.1	748.4	1861.4	12.3	100.0%	38.9%
b14_opt	1634.3	426.3	1624.2	6.4	99.4%	34.6%
b14_1	1668.3	372.5	1656.5	6.5	99.3%	36.9%
b14_1_opt	1534.4	269.0	1542.8	4.1	100.5%	38.2%
b15	2772.0	855.2	2799.2	12.6	101.0%	62.1%
b15_opt	2610.0	867.4	2605.2	12.0	99.8%	45.9%
b15_1	2799.4	1305.3	2787.8	21.9	99.6%	44.4%
b15_1_opt	2332.3	941.4	2340.2	13.6	100.3%	46.9%
b17	8546.8	9318.2	8561.4	142.4	100.2%	60.5%
b17_opt	8088.2	8896.2	8063.6	117.9	99.7%	54.7%
b17_1	8532.5	10831.6	8528.5	189.0	100.0%	50.6%
b17_1_opt	7192.4	8909.4	7212.8	121.2	100.3%	47.2%

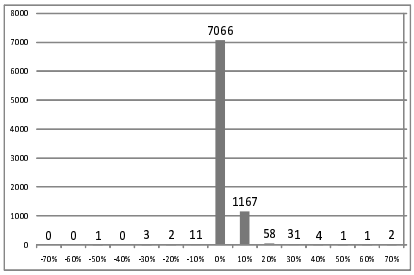

図 2: b15 の誤差の分布

4 まとめ

本稿では，ソフトエラー伝搬率の計算手法の評価を行った．今後の課題は，現実的な計算時間で実行でき，なおかつ妥当な値を計算できる手法の考案である．

謝辞

本研究の一部は，科学技術振興機構 (JST) の戦略的創造研究推進事業 (CREST)「統合的高信頼化設計のためのモデル化と検出・訂正・回復技術」の支援によるものである．

参考文献

[1] 上村大樹，戸坂清春，芹沢芳夫，岡秀樹，佐藤成生，"中性子ソフトエラーシミュレーションの新展開", 信学技報 Vol.105, No.2, pp.37-42, 2005.

[2] M. Abramovici, M. A. Breuer, A. D. Friedman, "Digital Systems Testing and Testable Design", IEEE Press, 1990.

[3] S. Krishnaswamy, S. M. Plaza, I. L. Markov, J. P. Hayes, "Enhancing Design Robustness with Reliability-aware Resynthesis and Logic Simulation", Proc. of ICCAD, pp149-154, 2007.